

Research Article

Temporal Convolutional Network with Wavelet Transform for Fall Detection

Xilin Lu ¹, Yuanxiang Ling ², and Shuzhi Liu ¹

¹School of Sports Science, Chongqing University, Chongqing 400000, China

²College of Communication Engineering, Jilin University, Changchun 130000, China

Correspondence should be addressed to Shuzhi Liu; cqutyliushuzhi@cqu.edu.cn

Received 8 April 2022; Revised 26 July 2022; Accepted 12 August 2022; Published 4 October 2022

Academic Editor: Ping Shi

Copyright © 2022 Xilin Lu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Fall detection is a challenging task for human activity recognition but is meaningful in health monitoring. However, for sensor-based fall prediction problems, using recurrent architectures such as recurrent neural network models to extract temporal features sometimes could not accurately capture global information. Therefore, an improved WTCN model is proposed in this research, in which the temporal convolutional network is combined with the wavelet transform. Firstly, we use the wavelet transform to process the one-dimensional time-domain signal into a two-dimensional time-frequency domain signal. This method helps us to process the raw signal data efficiently. Secondly, we design a temporal convolutional network model with ultralong memory referring to relevant convolutional architectures. It avoids the gradient disappearance and explosion problem usefully. In addition, this paper also conducts experiments comparing our WTCN model with typical recurrent architectures such as the long short-term memory network in conjunction with three datasets, UniMiB SHAR, SisFall, and UMAFall. The results show that WTCN outperforms other traditional methods, the accuracy of the proposed algorithm is up to 99.53%, and human fall behavior can be effectively recognized in real time.

1. Introduction

Human activity recognition (HAR) is a rapidly growing and promising branch of data science with many current applications, including healthcare surveillance [1, 2], smart home [3], and fall detection [4]. Among them, fall detection is one of the most important research topics in HAR. According to the World Health Organization (WHO) [5], falls are the second leading cause of accidental death worldwide. However, suppose this behavior is monitored and warned without delay. The time required for medical treatment can be significantly reduced, thus effectively reducing the potential risk of harm and death after a fall. Therefore, it is of great significance to propose models with high accuracy for identifying falling behaviors and applying them to suitable scenes and groups.

Wearable sensors are the basis for human behavior recognition systems, including fall detection [6]. At this stage, the methods for recognizing human falls are mainly divided into those based on signal and visual sensors. For signal sen-

sors, accelerometers, gyroscopes, and magnetometers can form an inertial measurement unit (IMU), where accelerometers detect linear motion and gravity by measuring acceleration in three axes (x, y, z); gyroscopes are used to measure rotation rates, including roll, yaw, and pitch. Moreover, with the development of camera techniques, such as the widespread use of GoPro, the practice of using wearable cameras for fall detection in the HAR field has increased over the last few years [7–10]. Sensors with image and video processing capabilities have been extensively investigated in this field [11, 12], and these approaches also differ significantly from signal-based sensor techniques. Fall detection based on visual sensors is not as widely used as signal-based sensors due to constraints, such as complex scenarios and the need to consider participants' privacy issues [13]. Therefore, despite their significance in HAR applications, this paper only plans to focus on signal-based sensors for fall behavior.

Machine learning and deep learning have brought disruptive changes to many fields in the past decade, including image recognition, target detection, speech recognition, and

natural language processing. As a typical behavioral recognition problem, many traditional machine learning and deep learning algorithms have solved sensor-based fall detection with good results. Models include support vector machine (SVM) [14], Google's deep neural network (DNN) [15], convolutional neural network (CNN) [16–19], long short-term memory network (LSTM) [20, 21], and recurrent neural network (RNN) [22–24]. However, when current fall detection based on signal sensor applies deep learning networks, especially using recursive architectures such as a single RNN model, it is sometimes challenging to capture global information of temporal features efficiently and accurately. Therefore, this paper proposes a new model, wavelet transform-temporal convolutional network (WTCN), to improve prediction accuracy.

Specifically, we build an improved WTCN fall detection system by using a lightweight temporal convolutional network (TCN) as the main structure and embedding the wavelet transform for the signal processing procedure. Thereinto, the wavelet transform helps us to process the raw signal data efficiently; the deep structure of TCN compensates for the lack of a single recurrent architecture with the advantages of stable gradients, flexible receptive field size, the low memory requirement for training, and variable-length inputs (Figure 1). In addition, this paper also uses a dropout layer to suppress the overfitting of the model and changes all the activation functions to PReLU. Finally, we apply different deep learning models (CNN, LSTM, CNN+LSTM, TCN, and WTCN) to the datasets (including UniMiB SHAR, SisFall, and UMAFall) reorganized by our research team for experiments and compare their fall detection performances among them. The results show that WTCN outperforms the baseline recursive architecture in all four aspects of the loss function, accuracy, recall, and precision.

To summarize, the main contributions of this paper are as follows:

- (1) In terms of datasets, our team reorganized a wide range of publicly available datasets on human activity that included falling behavior, involved UMAFall, SisFall, and UniMiB SHAR. Moreover, we relabeled the behaviors of daily living (DAL) and falls (FALL) among these three datasets and cleaned the redundant and invalid data
- (2) About data processing, the wavelet transform method used in this paper can improve the predictive capability of our model to a certain extent. Specifically, the 2D images, transformed from the 1D data, contain both time and frequency domains, thus giving a complete picture of the signal characteristics. This procedure also provides the basis for subsequent improvements in the accuracy of recognizing fall behaviors
- (3) In the model structuring section, a new model WTCN is proposed to improve the efficiency of fall detection in this paper. To test the recognition effectiveness of this model, we have compared it with the

CNN, LSTM, CNN+LSTM, and TCN networks based on the baseline, and the integrated dataset mentioned above is applied. The experimental results demonstrate that our proposed model has better performances with higher recognition and classification accuracy, which could also provide suggestive ideas for subsequent research

- (4) As a whole, no other researcher has been found to use the integrated TCN model for fall detection, so this paper supplements this and demonstrates that it is a relatively great model with well-performed results in this field

The rest of this article is described below. Section 2 illustrates the development and overview of related works involved in fall detection algorithms, including an introduction to deep learning networks, and our preparation for improving models. Section 3 shows the whole framework of our WTCN model, which includes casual convolutions and dilated convolutions, TCN network components, and wavelet transformation. Section 4 describes the initial situation of the three datasets and the process of pre-processing procedure performed by our team. Besides, this part also adds some details related to the training process, and experimental settings. Section 5 shows and discusses our experimental results. Finally, Section 6 concludes with a summary of the main work in this paper.

2. Related Work

At this stage, machine learning and deep learning algorithms are widely used in HAR. The increased public datasets, hardware acceleration capabilities, and algorithmic advances have provided a solid foundation for researchers to develop models with excellent performance and sophistication. This section will describe algorithms applied to the field of fall detection.

Machine learning algorithms have recognition classification capabilities that automatically learn data attributes and build classification models. If fall detection is considered a typical classification problem, based on a training set consisting of fall and non-fall data, typical machine learning algorithms, such as SVM [14, 25, 26], DNN [15], boosted decision tree (BDT), artificial neural network (ANN) [27], and k -nearest neighbor (k -NN) [28], can be used to construct fall detection models [29–31]. Mrozek et al. [32] proposed scalable system architecture for remote monitoring of fall behavior in an elderly population in which the applicability of several machine learning algorithms to the detection process was evaluated. Specifically, the researchers validated random forest (RF), ANN, SVM, and BDT classifiers, with BDT performing the best, achieving an average accuracy of 99% in the SisFall dataset. However, feature selection is the key to the success or failure of machine learning algorithms, and the accuracy of fall detection can be significantly affected if the manually extracted features are not ideal.

Compared to machine learning-based fall detection algorithms, deep learning algorithms can select features

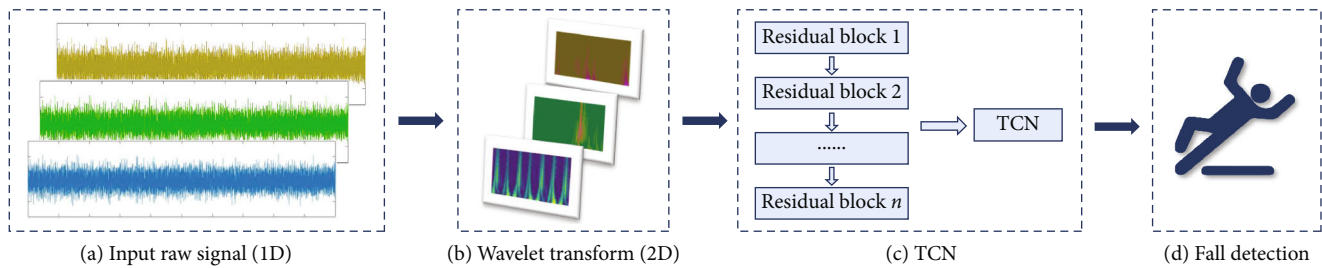


FIGURE 1: The workflow of WTCN. (a) The first step to input raw signals from sensor-based datasets. Then, through wavelet transform (step b), 1D signals could be converted to 2D images. (c) About model structuring and training procedures based on an optimized TCN network. Finally, we can detect the fall behaviors through the above steps.

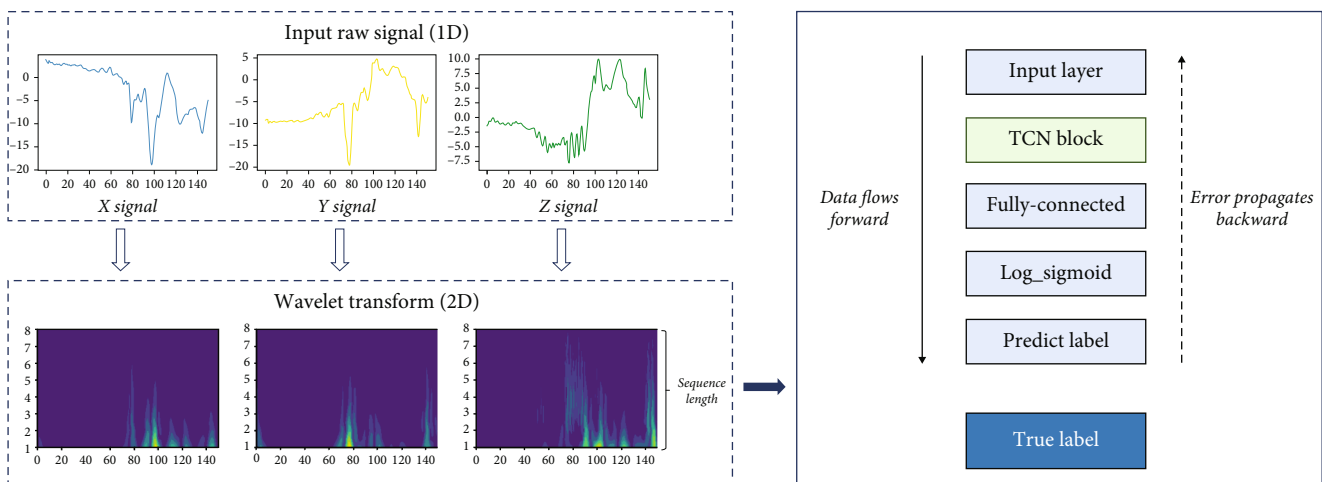


FIGURE 2: The diagram of WTCN network structure.

autonomously and have powerful learning capabilities. At present, there exist some kinds of deep learning algorithms that have shown their ability to capture local features and have achieved remarkable performance in the field of fall detection, such as CNN, RNN, and LSTM. Specifically, in contrast to fully connected neural networks, the pyramidal structure of CNN enables them to aggregate low-level local features into high-level semantic structures, which allows them to learn further about superior features. There are several mechanisms for CNN-based time series classification problems, which can be divided into two categories. The first of these would use time-series data as input to a 1D grid. For example, Zheng et al. [33, 34] separate multivariate time series into univariate time series and then perform feature learning on each univariate series separately. The second category would convert the 1D time-series data into 2D image features, which would be subsequently processed. For instance, some researchers have attempted to encode time series data into two-dimensional images using a short-time Fourier transform as input to a CNN [35, 36]. These studies also provide references for processing raw sensor signals in this paper.

Since it was proposed in 1991 [37], RNN with time series as input has been widely used for human activity classification or gesture estimation [38–44]. Many researchers have carried out extensive work to improve the performance of

RNN models in HAR [45–47], and Torti et al. [48] propose an RNN system for fall detection suitable for a microcontroller embedded implementation, with an overall detection rate of 98%. It is worth noting that the time processing ratio of its input signal can reach 0.3, demonstrating the feasibility of the proposed model for real-time remote monitoring. Some scholars have designed various RNN-based models, including IndRNN [49], CTRNN [50], PerRNN [51], and CBO-RNN [52].

Fall detection tasks would perform better when a model is set up with longer contextual information and time intervals. However, this can lead to gradient disappearance or explosion problems when backpropagation [53] is performed. LSTM [54] has been introduced to address these challenges. Notably, LSTM has been shown to solve the long-term dependency problem in RNN, and previous studies have demonstrated the high performance of LSTM in HAR [55, 56]. Researchers have also explored other architectures related to LSTM to improve the baseline of HAR datasets. For example, Hu et al. [57] proposed a loss function, and Zebin et al. [51] combined LSTM with batch normalization to achieve 92% prediction accuracy on raw accelerometer and gyroscope data, while Ordóñez and Roggen [58] proposed a hybrid CNN and LSTM model (DeepConvLSTM) for activity recognition based on data from multimodal wearable sensors. In addition,

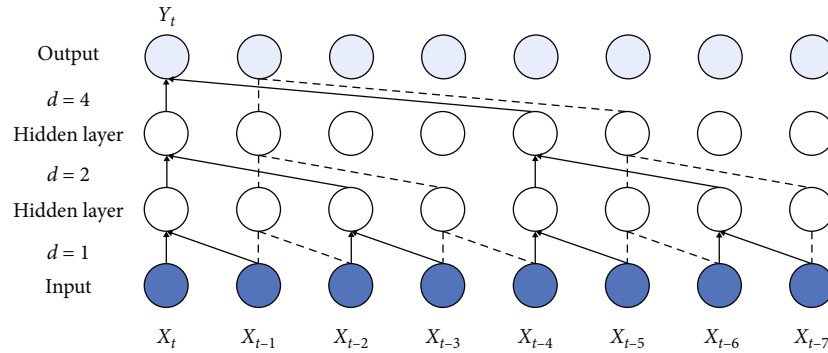


FIGURE 3: A dilated causal convolutions with dilation factors $d = 1, 2,$ and 4 and filter size $k = 2$.

other researchers have developed CNN-LSTM models for different application scenarios by combining the feature extraction capability of CNN and the time-series data processing capability of LSTM [59–66].

These studies demonstrate the potential of deep learning network models applied to the field of fall detection. Many detection algorithms can achieve high accuracy and successfully extract the user’s activity state from sensor data. With in-depth study, we find that using a single algorithm has limitations, and it is difficult to adapt to the changes in human falling behavior in various scenarios. In contrast, the hybrid algorithm shows substantial superiority. Combining the advantages of different algorithms, it can better deal with the multienvironment and multipurpose tasks of fall classification problems. Moreover, by observing the accelerometer signals of ADLs and falls in databases, we have found that the duration of the falling motion is relatively short, which means that the frequency domain is more informative than the time domain. Specifically, some of the actions of ADLs and falls are similar when analyzing the time domain, such as lying down from standing and falling backward. However, it becomes relatively easy to distinguish after converting these actions into signal waveforms of the frequency domain. Therefore, to further improve the recognition accuracy of fall detection, combining previous studies and our observations, this paper adopts a wavelet transform method to process the raw sensor signal data. It can maximize the retention of information links and temporal features of the actions before and after the fall.

As previously explained in this section, researchers have used classic deep learning algorithms to achieve a state where the network has a memory for prior information, such as using RNN alone or hybrid applying CNN+LSTM. However, to varying degrees, the models in these studies suffer from slow running times, inflexible sensory domains, gradient disappearance and explosion, and high memory usage. In summary, a CNN-based improved TCN network model has been selected for training to achieve our optimization objective. That is, the new proposed model can satisfy the automatic extraction of the signal feature. At the same time, the network has a memory for the prior sequence information, thus helping to make efficient and accurate decisions about fall detection. Specifically, the improvements

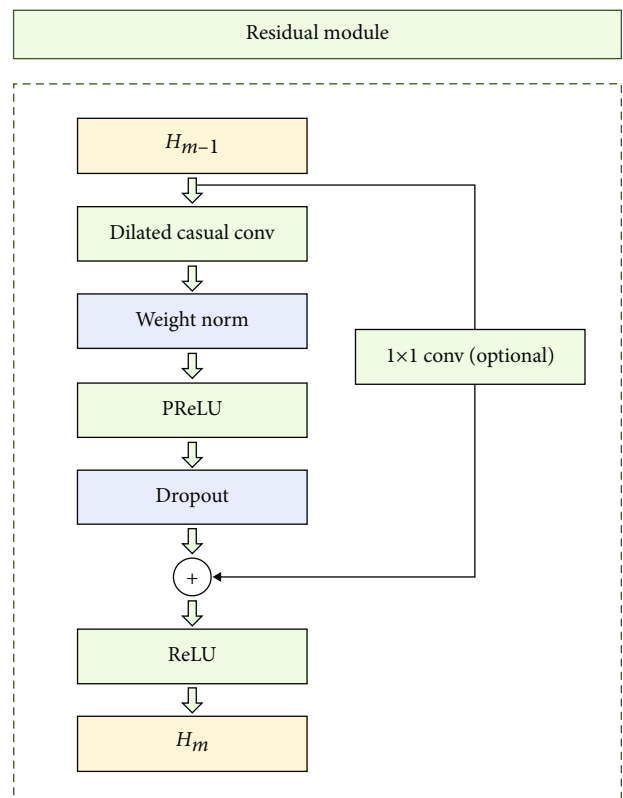


FIGURE 4: The structure of the residual module.

of our model compared with other studies are outlined as follows:

- (i) There is an optimization of the running time. Given a time series signal, our model allows the network to map the input directly to get the result without requiring sequential processing like RNN, which cannot be parallelized
- (ii) It can perform stable gradient descent. Compared with the gradient disappearance and explosion problems that often exist in RNN, the residual network included in the TCN model in this paper can solve them to a certain extent

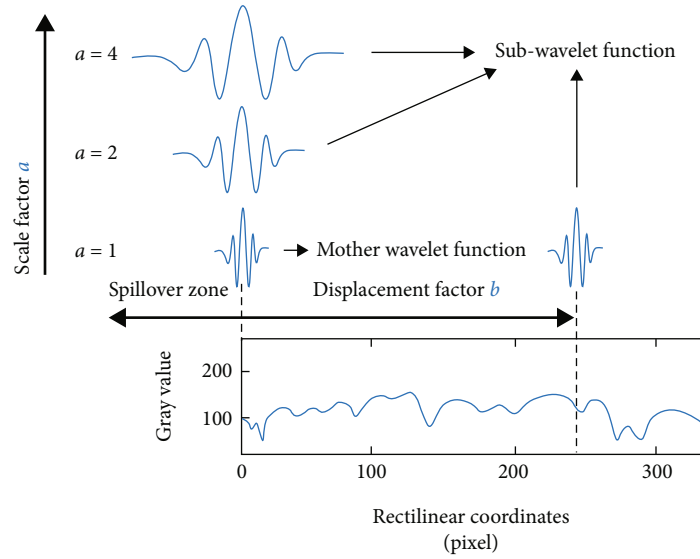


FIGURE 5: The calculation process of 1D-CWT.

- (iii) It has a lower memory footprint. With the same number of layers, our model runs with less memory due to the sharing of convolutional kernels within a layer, compared to RNN, which saves information at each step

3. Framework

First, we briefly describe the main structure of our WTCN model, which includes casual convolutions and dilated convolutions. Then, we give a brief account of the network components used in this paper, such as weight norm, residual block, and dropout. Third, we introduce the wavelet transform with which we update the 3D acceleration sensor signals. Finally, the result of serially applied wavelet transform and TCN are called the WTCN. The overall architecture is shown in Figure 2.

3.1. Dilated Casual Convolutions. The input data used for fall detection in this article is time series data acquired by accelerometers set at a certain sampling rate. Therefore, before introducing the dilated casual convolutions network, we first introduce the nature of the sequence modeling task. We use the acceleration sensor to obtain the electrical signal sequence x_0, x_1, \dots, x_T as the function's input after the analog-to-digital conversion procedure, and hope to predict the corresponding fall detection results \hat{y}_T . The key constraint is that we only use previously observed input to predict \hat{y}_T . In a mathematical form, the sequence modeling network is the function $f(\cdot)$ that generates the above mapping, satisfying $\hat{y}_T = f(x_0, x_1, \dots, x_T)$ [67]. Then, the sequence modeling task's goal is to find a network $f(\cdot)$ that is trained to minimise the loss function of its predicted and actual results.

To model sequences, we need to deal with variable-length sequences, keep track of long-term dependencies, maintain order information, and share parameters within sequences. RNNs meet these sequence modeling design cri-

teria and are considered a common model for handling sequence modeling tasks. However, TCNs outperform RNNs on specific tasks and datasets, such as Seq, MNIST, and Music Nottingham [67].

About the characteristics of TCNs, firstly, this network can take an input sequence of any length and map it to an output sequence of the same length. Secondly, because the convolutional layer of the TCN network adopts causal convolution, it can avoid the data leakage problem from the future into the past [68, 69]. However, when dealing with tasks with long sequence lengths, in order to obtain adequate history information, the network depth or the convolution kernel size would increase as the TCN input sequence length increases. As a result, gradient explosion or gradient disappearance is more likely to occur in the training process. Therefore, dilated convolutions are introduced to solve this major shortcoming. This structure sets a fixed step size for each two adjacent convolution kernels, which enable an exponentially larger receptive field [70]. Using larger dilation enables an output at the top level to represent a wider range of inputs. Thus, receptive field can cover all input sequence data. Therefore, the dilated causal convolution structure is shown in Figure 3, in which there are 3 dilated causal convolutions with dilation factors $d = 1, 2,$ and 4 and filter size 2. Output Y_t 's receptive field can reach X_{t-7} .

Next is the specific design of this paper for dilated casual convolutions. First, we specify the receptive field. Specifically, assuming that the kernel size is k_s and the dilated causal convolution layer number is L_n , dilation is set to 2^{L_n-1} for each layer, the receptive field of the first layer network can reach k_s series length, the receptive field of the second layer network can reach $3k_s - 2$, and the receptive field of the Nth layer network can reach $T = 1 + (k_s - 1) * (2^{L_n} - 1)$ series length. Second, in order to extend dilated casual convolutions with receptive fields up to the sequence length L_s of the input signal, the network needs to satisfy the condition $T \geq L_s$. Thus, the formula that the size of the

TABLE 1: Overall feature of three datasets.

Datasets	ADLs	Falls	Data source	Sampling rates	Location	Subject numbers	Age	Gender	
								Female	Male
UniMiB SHAR	Yes	Yes	A Samsung phone with a BMA220 acceleration sensor	50 Hz	Trouser pockets	30	18 - 60	24	6
SisFall	Yes	Yes	A self-developed embedded device composed of a CPU, a ADXL345 accelerometer	200 Hz	Waist	38	19 - 75	19	19
UMAFall	Yes	Yes	An android smartphone and a set of mobility sensors attached to different parts of the body	200 Hz	Chest, waist, wrist, ankle	17	14 - 55	6	11

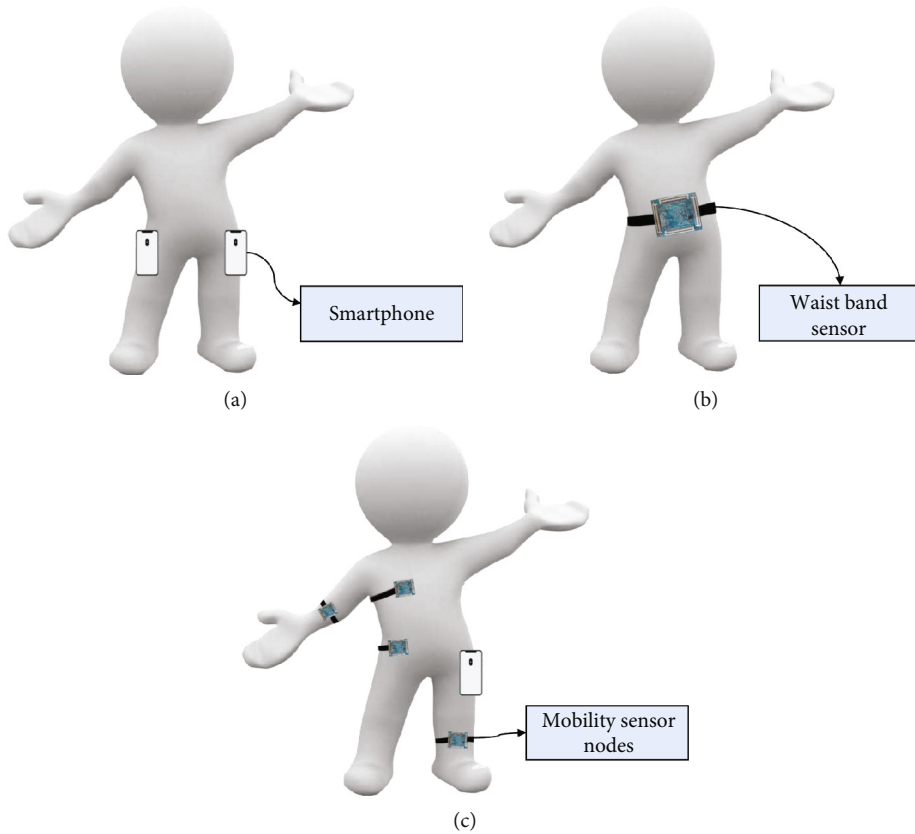


FIGURE 6: Sensor location from (a) UniMib-SHAR, (b) SisFall, and (c) UMAFall datasets.

convolution kernel and the number of convolution layers need to be satisfied can be expressed as

$$1 + (k_s - 1) * (2^{L_n} - 1) \geq L_s. \quad (1)$$

3.2. Residual Connections. Residual connections demonstrate the benefits of using additive merging signals in image recognition, particularly object detection [71]. Some researchers believe residual connections are essential for training deep architectures [71–73]. As the TCN receptive field is more dependent on the convolutional depth, kernel size, and dilation of the network, problems such as network degradation may arise as the depth of the network increases.

Therefore, in order to ensure that our network can be trained effectively and stably, the TCN model in this paper introduces the residual module instead of the traditional convolutional layer structure, which is presented in Figure 4.

Specifically, for this network structure, there is one layer of dilated causal convolution and a nonlinear activation function ReLU in this block. For normalization, weight normalization to the convolution kernel is used. In addition, we also add dropout regularization after the dilated causal convolution, thus avoiding the overfitting problem to some extent. In later experimental sessions, we tried adding more convolutional layers or modifying the activation function to explore the best block design.

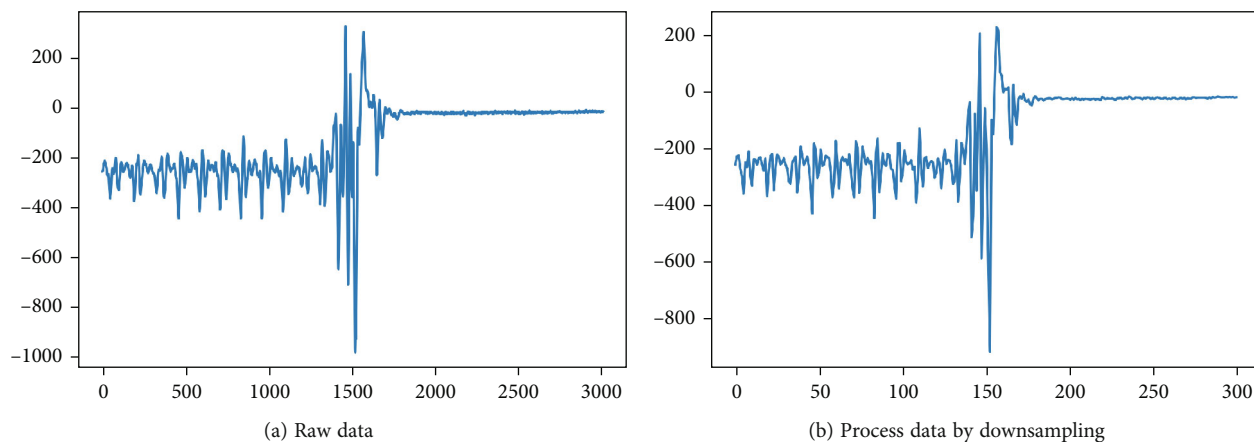


FIGURE 7: The process of downsampling signal data.

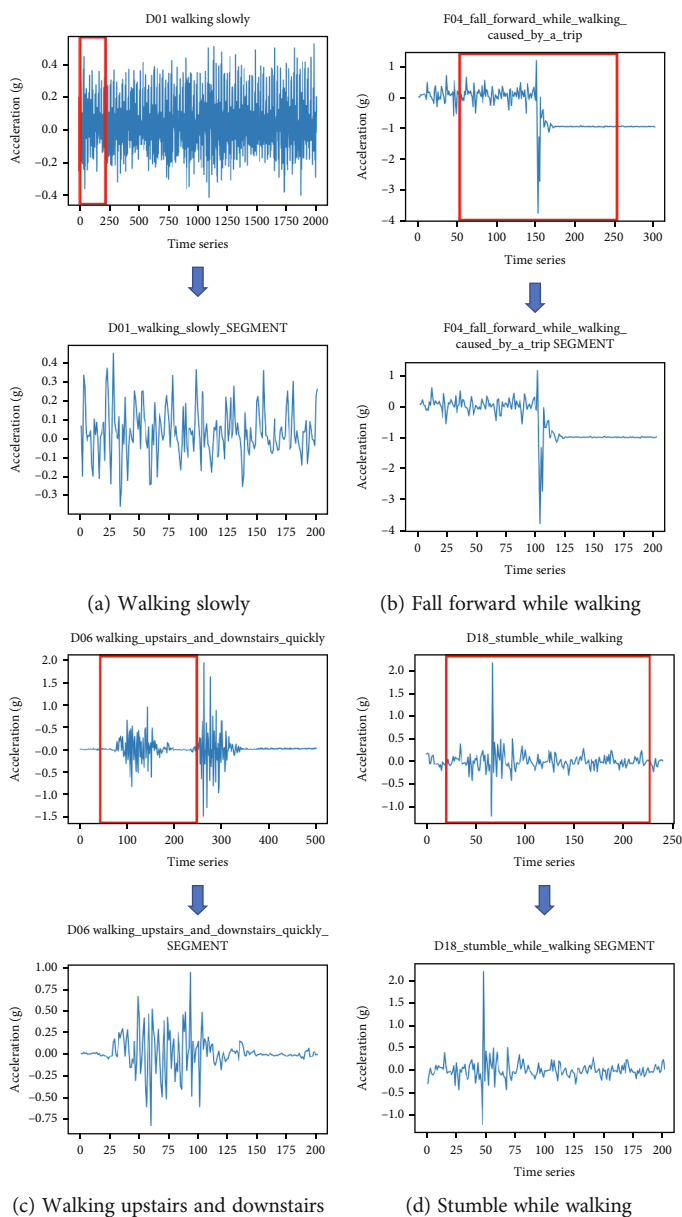


FIGURE 8: Some visualization samples about segmentation procedure.

TABLE 2: Parameter configuration of the proposed network.

Layer	Parameters
1 wavelet transform	Mother wavelet : Mexican hat, sampling_rate = 16, totals cale = 8
2 TCN block	Kernel size = 9, kernel number = 16, dilation rate = 1
3 TCN block	Kernel size = 9, kernel number = 16, dilation rate = 2
4 TCN block	Kernel size = 9, kernel number = 16, dilation rate = 4
5 TCN block	Kernel size = 9, kernel number = 16, dilation rate = 8
6 TCN block	Kernel size = 9, kernel number = 16, dilation rate = 16
7 TCN block	Kernel size = 9, kernel number = 16, dilation rate = 32
8 fully-connected	Activation = log-sigmoid

3.3. *Wavelet Transform.* Oftentimes, the information that cannot be readily seen in the time domain can be seen in the frequency domain. Currently, there are two the most frequently used ways to convert the time domain into the frequency domain: Fourier transform and wavelet transform [74]. However, there is no temporal information available in the Fourier transform signal. When analyzing fall detection data, we are more interested in what spectral component occurs at what time interval. Consequently, the wavelet transform is much more suitable for analyzing time-frequency representation in this paper.

The wavelet transform is a mathematical method of spectral analysis developed based on the Fourier transform [75, 76]. It can be automatically adapted to the requirements of time-frequency signal analysis by “stretching” and “translating,” so that it can focus on arbitrary details of the signal [77]. Wavelet transform methods can be divided into discrete wavelet transforms (DWT) and continuous wavelet transforms (CWT) [78]. Depending on the spatial dimension of the signal to be analyzed, the continuous wavelet transform can take different forms, such as one-dimensional and two-dimensional [79]. This paper plans to use the one-dimensional continuous wavelet transform for further study.

The mathematical process of the one-dimensional continuous wavelet transform can be described as follows: firstly, obtaining a series of subwavelet functions by stretching and translating the mother wavelet function; secondly, doing convolution with the unprocessed signal; finally, getting a set of wavelet coefficient matrices (as shown in Figure 5).

After decades of development, scholars have proposed a variety of wavelet functions, including Haar [80], Morlet [81], Daubechies [82], Coiflets [83], MexicanHat [84], and other wavelets. Each wavelet has different properties such as support length, filter length, and center frequency. We can choose the proper wavelet function according to the actual processing requirements for research.

4. Experiment

4.1. *Fall Detection Tasks.* We evaluated CNNs, RNNs, and WTCNs on datasets commonly used to benchmark fall detection tasks. The datasets applied to fall detection tasks

TABLE 3: Confusion matrix.

Predict	Positive	Negative
True	True Positive (TP)	True Negative (TN)
False	False Positive (FP)	False Negative (FN)

consist of two main types: vision-based datasets and sensor-based datasets. Examples of vision-based datasets are KTH [85] and Wieszmann [86]. Sensor-based datasets include four types: object sensors, wearable sensors, hybrid sensors, and ambient sensors. Vankastern Benchmark [87] and Ambient kitchen [88] are object sensor-based datasets, UCI-HAR and WISDM [89] are wearable sensor-based datasets, and Opportunity [89] is a hybrid sensor-based dataset, and AAL [90] is an ambient sensor-based dataset. This paper mainly focuses on datasets based on object sensors (mainly smartphones), specifically UniMiB SHAR [14], and wearable sensor types, SisFall [91], and UMAFall [92]. Table 1 shows basic information about these three datasets, including the type of sensor, frequency of collected signals, age, and gender of the subjects. Moreover, Figure 6 shows further information about the sensors’ location from the experimental subjects using visualization images.

4.1.1. *Object Sensor Dataset.* The UniMiB-SHAR dataset [14] is acquired with an Android smartphone Application from 30 subjects (6 male and 24 female) for human activity recognition and fall detection. The dataset is sampled at a frequency of 50 Hz using the 3D accelerometer of a Samsung smartphone, including 11,771 samples of both human activities and falls.

In this dataset, each accelerometer signal is segmented into a window of almost 3 seconds each time (151 samples) around a peak of the accelerometer signal is located at time t when the magnitude of this signal is high than $1.5g$ (with g being the gravitational acceleration). The magnitude at the preview time $t - 1$ was lower than 0. In addition, this is a publicly available dataset that many researchers have used to train and test their models directly [93–95]. We have also included all the data from this dataset.

TABLE 4: Settings of the models used on the UniMiB-SHAR and SisFall-UMAFall datasets in the comparison experiments.

UniMiB-SHAR ¹ and SisFall-UMAFall ²		
Model	Parameter	Value
CNN	Convolutional kernel size for blocks 1	(32, 3) ¹ , (51, 3) ²
	Convolutional sliding stride for block 1	(1, 1) ¹ , (1, 1) ²
	Convolutional kernels for block 1	100 ¹ , 60 ²
	Pooling sizes for block 1	(2, 1)
	Neurons in fully connected layer	6000 ¹ , 9000 ²
LSTM	LSTM cells in layers 1 and 2	151 ¹ , 151 ¹ , 200 ² , 200 ²
	Output dimensions of LSTM cells in layers 1 and 2	1000 ¹ , 1000 ¹ , 1200 ² , 1200 ²
	Neurons in fully connected layer	1000 ¹ , 1200 ²
CNN-LSTM	Convolutional kernel size for blocks 1	(32, 3) ¹ , (51, 3) ²
	Convolutional sliding stride for block 1	(1, 1) ¹ , (1, 1) ²
	Convolutional kernels for block 1	100 ¹ , 60 ²
	Pooling sizes for block 1	(2, 1)
	LSTM cells in layers 1 and 2	60 ¹ , 60 ¹ , 150 ² , 150 ²
TCN	Output dimensions of LSTM cells in layers 1 and 2	1000 ¹ , 1000 ¹ , 1200 ² , 1200 ²
	Neurons in fully connected layer	1000 ¹ , 1200 ²
	1 TCN block	(5, 16), $d = 1$
	2 TCN block	(5, 16), $d = 2$
	3 TCN block	(5, 16), $d = 4$
	4 TCN block	(5, 16), $d = 8$
	6 TCN block	(5, 16), $d = 16$
	7 TCN block	(5, 16), $d = 32$
	Neurons in fully connected layer	16

¹, ²The parameters of CNN, LSTM, and CNN-LSTM models used in the UniMiB-SHAR, and SisFall-UMAFall, respectively.

4.1.2. *Wearable Sensor Datasets.* SisFall is a publicly available dataset containing records of human activities of daily living and falls [91]. Unlike most datasets based on smartphones [96, 97] to collect data, it uses a dedicated custom sensing device. In this dataset, data was sourced from two triaxial accelerometers (ADXL345 and MMA8451Q) and a triaxial gyroscope (ITG3200). Moreover, the sampling frequency is 200 Hz, and the acquisition site is the waistband of experimental subjects.

Besides, the UMA-Fall dataset [92] includes 746 samples from various test subjects. The experimental data were collected from five wireless sensors placed on the subjects, including a smartphone and four sensors. Regarding these five sensors' location, the smartphone was placed in the subject's pocket, and the four sensors were worn on the subjects' ankles, wrist, chest, and waist, respectively. All five sensors could transmit triaxial accelerometers, triaxial gyroscope, and magnetometer data via Bluetooth.

4.1.3. *Integration Standard of Wearable Sensor Datasets.* SisFall and UMAFall are wearable sensor-based datasets, which both have long time series signals compared to UniMiB SHAR. As mentioned earlier, using non-categorical sample data from different body locations can significantly reduce the accuracy of predictive models. As UMAFall was collected from five different locations, our team has sorted and

included only its signals from waist sensors in our integrated database. It is worth noting that the sampling frequency of SisFall and UMAFall databases are different, with 200 Hz and 20 Hz, respectively (in this case, we only consider the waist wearable sensors). Therefore, we first need to down-sample the signal data of SisFall from 200 Hz to 20 Hz (Figure 7), thus enabling us to analyze the fall movements further using the settled data with the same sampling frequency subsequently.

After downsampling, we need to do the segmentation procedure (Figure 8, the marked red fields in the figures are the split windows).

Firstly, according to the characteristics of time series signal among fall behaviors in SisFall Dataset, regions with large rates of change ($>1.5g$, g is gravitational acceleration) in gravitational acceleration data have all the characteristics of processes occurring before and after fall behavior. Besides, the sequence lengths of the object sensor datasets and wearable sensor datasets are inconsistent, with the former having a fixed sequence length of 151 and the latter having a sequence length of up to 2000 for the ADLs and 300 for the Falls. Since we wanted to verify the performance of our WTCN model with low-frequency wearable sensors and to ensure that the data processed by the wearable sensors and the UniMiB-SHAR dataset input sequence were of similar length, a signal window of 10s is chosen was chosen to intercept the data.

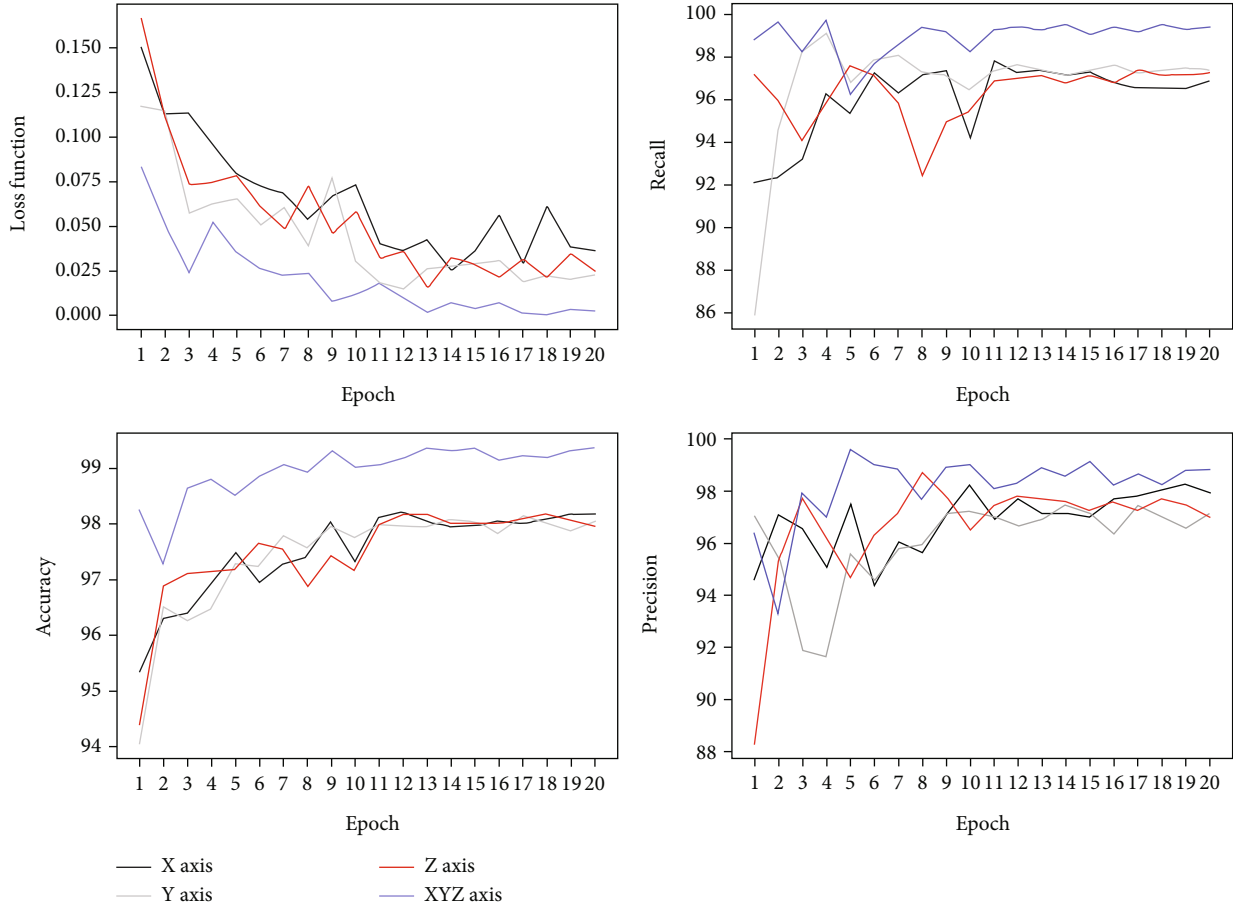


FIGURE 9: Loss function, recall, accuracy, and precision of the number of epochs for 1- and 3-axis neural networks. For one dimension, the black, grey, and red lines show the training performance of the X-, Y-, and Z-axis data input to the model; for three dimensions; the blue line shows the training performance of the XYZ axis acceleration data input to the model.

Secondly, the datasets needed to be further segmented according to the trial length of different action types. Specifically, because of the different lengths of each trial for ADLs and falls, we segmented the data using a 10s long signal window, based on which the time series data for different action types were divided into different groups. We first segmented four ADL types of the time series data (D01 walking slowly, D02 walking quickly, D03 jogging slowly, and D04 jogging quickly), finally divided into ten groups. Secondly, since there are three ADLs' trial lengths up to 25s (D05 walking upstairs and downstairs slowly, D06 walking upstairs and downstairs quickly, D17 standing, getting into a car, remaining seated, and getting out of the car), we split them into two groups. Besides, the rest of the behavior types were grouped. At this point, we have processed the data to obtain window samples with a time series length of 200 (20 Hz sampling rate, 10 s length).

In addition, we also analyzed the experimental objects from the original baseline databases and excluded irrelevant data. Specifically, in SisFall database, only SE06 (an older person with a high level of health) performed the falls experiment, while the rest of the elderly only did ADLs. Therefore, to ensure consistent data proportions

for both the ADLs and fall action labels, we excluded data from the ADL sample for the rest of the elderly group except for SE06 and included the data of SE06 in our integrated database.

4.2. Model Settings. Regarding model settings for the entire network structure, the parameters of each layer are shown in Table 2. The network source is a three-channel acceleration sensor, and each channel is a 1D sequence of data with length H . The input data flows over Wavelet Transforms blocks, 6 stacked TCN blocks, and a fully connected (FC) layer (with `log_softmax`). The network's output is compared with the fall or ADL label, and then the error is backpropagated to update the network. In this current work, though there are many widely-used CWT mother wavelets, we only select several of them: Morlet wavelet, Mexican hat wavelet, and Gaussian.

Furthermore, based on the block in residual connections mentioned in Section 3.2, further modifications have been attempted, such as the proposed replacement of the ReLU layer with PReLU. To be specific, the PReLU (Parametric Rectified Linear Unit) is an activation function, which sacrifices hard-zero sparsity for a gradient and thus is more robust during optimization [98]. This function is

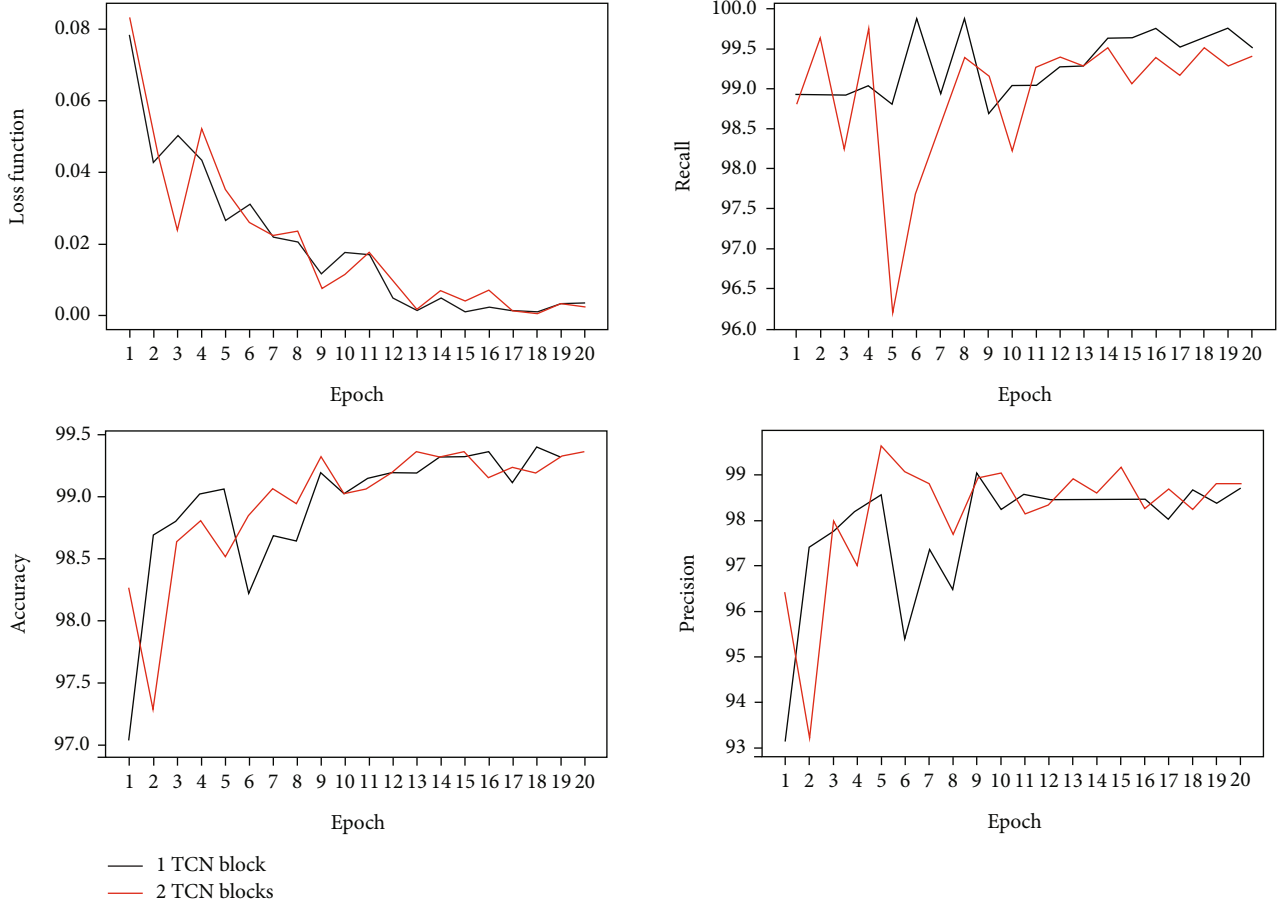


FIGURE 10: Loss function, recall, accuracy, and precision of number of epoches for 1 and 2 TCN Blocks neural networks.

shown as Equation (2) (i indicates a different channel), where a_i is the parametric rate which is updated by following Equation (3).

$$\text{PReLU}(x_i) = f(x) = \begin{cases} x_i, & x > 0, \\ a_i x_i, & x \leq 0, \end{cases} \quad (2)$$

$$\Delta a_i := \mu \Delta a_i + \epsilon \frac{\partial \epsilon}{\partial a_i}. \quad (3)$$

4.3. Performance Evaluation and Comparison. This article uses NLLLOSS (the negative log-likelihood loss) as a loss function during model training. Besides, it is helpful to train a classification problem with two classes (see Table 3). Moreover, we take accuracy, precision, and recall as model evaluation indicators when model testing.

$$(1) \text{ Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN})$$

$$(2) \text{ Precision} = \text{TP} / (\text{TP} + \text{FP})$$

$$(3) \text{ Recall} = \text{TP} / (\text{TP} + \text{FN})$$

4.4. Model Training. Our deep-learning models were implemented using the PyTorch library. The computing platform was equipped with an AMD Ryzen 5 3500X 6-Core Proces-

sor at 3.59 GHz, 16.0 GB RAM, and a 6 GB NVIDIA GeForce GTX 1660 SUPER GPU. All parameters of the models were randomly orthogonally initialized, and an Adam optimizer was adopted for back-propagation learning when model training. The batch size and epochs are 32 and 30. The initial learning rate is $2e-3$ and will be smaller every 10 epochs.

4.5. Comparison with the State of the Art. In this section, we will present the state-of-the-art model settings used for the comparison experiments.

The state-of-the-art CNN generally comprises an input layer including 3D accelerate data, one convolutional layer followed by nonlinear and pooling layers, and one fully connected layer. In the convolution layer, we apply many 1D convolution kernels over an input signal composed of several input planes. Besides, the convolution kernels automatically learn local and short-term features in the time domain. After the activation and max pooling layers, the feature maps will be flattened and passed through one fully connected layer. Finally, the probability of each class will be computed by a softmax layer.

An LSTM can accurately memorize the valid information from the new input in the time domain, and forget the long-term memory information it no longer needs. First, we input three-dimensional electrical signals

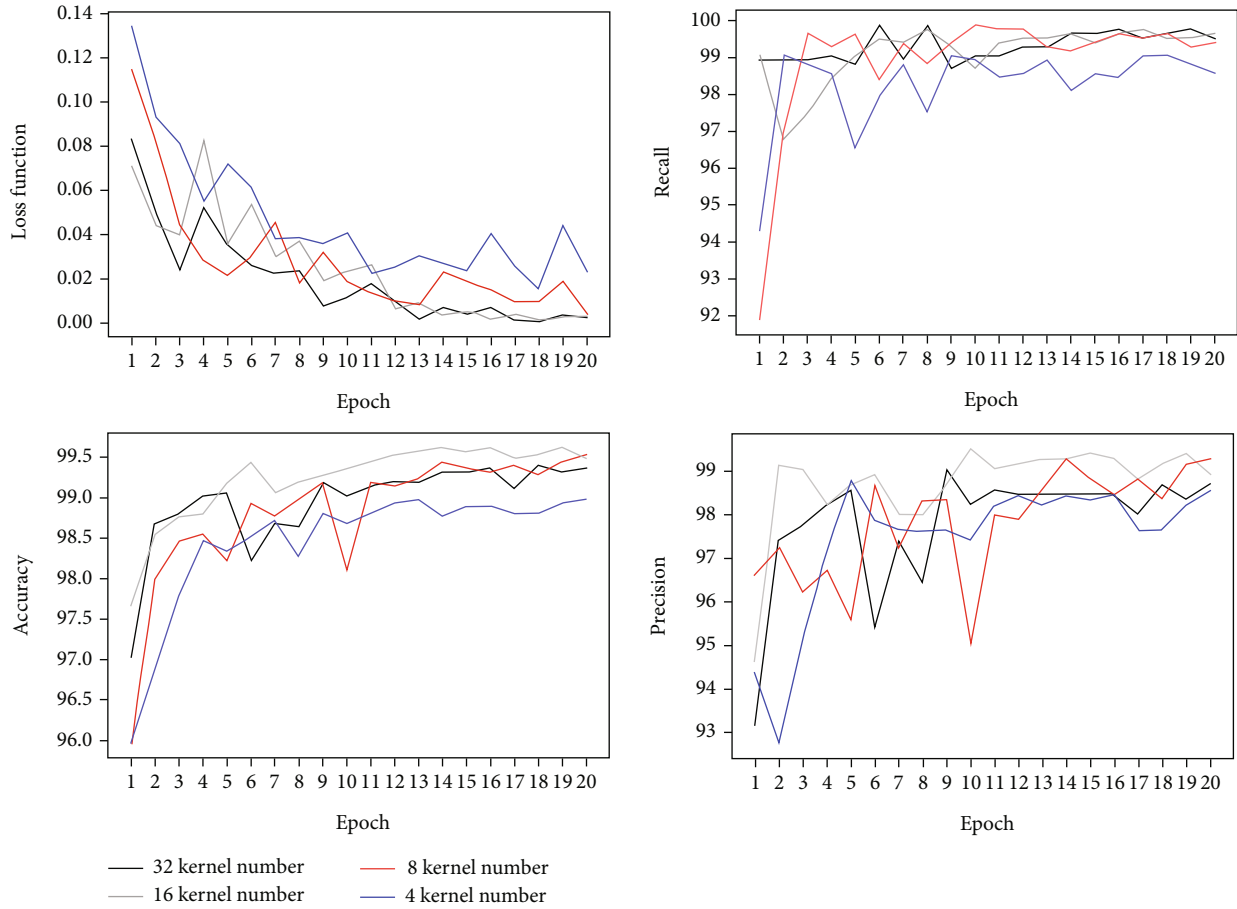


FIGURE 11: Loss function, recall, accuracy, and precision of number of epoches for 32, 16, 8, and 4 kernel numbers in each conv1D layer of TCN Blocks neural networks.

with the same sequence length to the LSTM network. Then, it inputs all the hidden nodes of the latest sequence into the fully connected network after acquiring longer memory. Finally, the classification probabilities predicted by the model are obtained through the softmax layer.

A hybrid convolutional and recurrent network structure is often used for benchmarking one-dimensional signal data. Similarly, this paper builds our hybrid (CNN-LSTM) model. Specifically, we first input signal data to obtain time-domain feature maps through the convolutional layer. Subsequently, we enter it into the LSTM network to learn long-term time-dependent information. As a result, the classification results can be obtained after LSTM followed by a fully connected and softmax layer.

In addition, we have designed control experiments based on TCN. Table 4 shows the details of parameter settings for designing the state-of-the-art model.

5. Results and Discussion

5.1. Different Axes: $x/y/z$ -Axes. This section needs to identify the dimensions of the acceleration data axes in which the model performs best. Firstly, we input the X-, Y-, and Z-axes of the acceleration sensor data, respectively, into the

model and train it. We found that the four metrics for evaluating the model's performance were similar when a single-dimensional acceleration axis was input, thus indicating that the model's sensitivity to a single-dimensional input data source was relatively consistent. To investigate the model's sensitivity to the data source, we used three-axis acceleration data for comparison experiments. As shown in Figure 9, we found that the three-axis acceleration input data can achieve better performance on the WTCN model compared to the one-dimensional input, and its accuracy can reach 99.36%.

5.2. Block Trial-1 Deep Blocks. In this section, we tried to optimize the network by adjusting the depth of the TCN blocks. As mentioned earlier, we assumed in the model construction phase that the performance of the TCN block is best when the residual matrix in the TCN block is one layer. To test this hypothesis, we first increased the number of convolutional layers in the TCN block to 2 and conducted a comparison experiment. The results showed that the four evaluation metrics (loss function, recall, accuracy, and precision) of the TCN block with two layers of residual matrix fluctuated more during the training process than the TCN block with one layer (see Figure 10). This performance is not conducive to faster convergence of the model. Therefore, we decided to use a 1-layer residual matrix in the TCN block

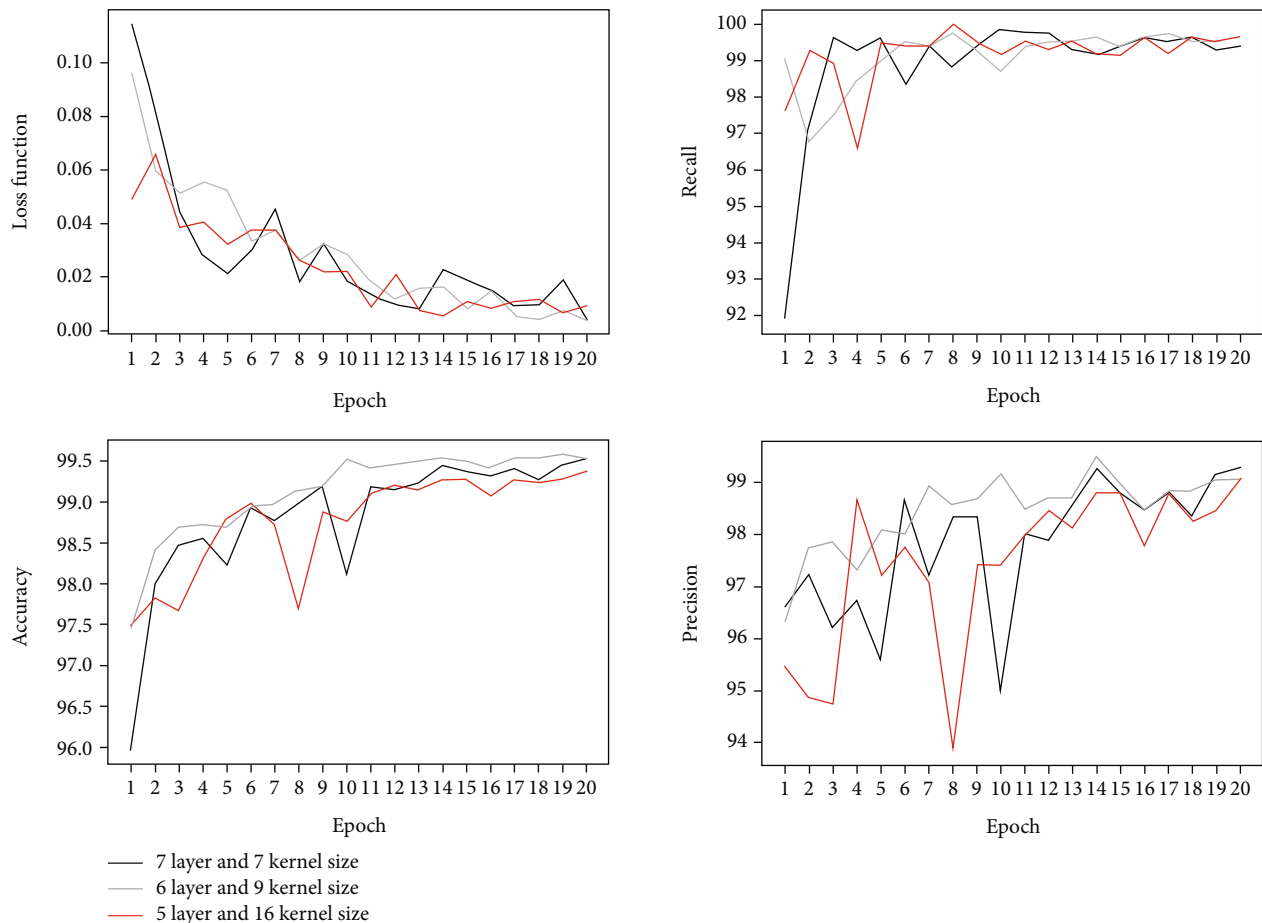


FIGURE 12: Loss function, recall, accuracy, and precision of number of epoches for different conv1D layers and kernel sizes of TCN Blocks neural networks.

to better fit the results, which is more time- and resource-efficient.

5.3. Network Variations (Kernel Number/Layer Size/Kernel Size). We should first find the optimal number of convolutional kernels for the model for network variations. Based on the experimental results in Section 5.1 (a TCN block structure with one layer of convolutional depth), we experiment with changing the kernel number. As mentioned earlier, we assumed in the model construction phase that the performance is best when the number of convolutional kernels is 16. In order to test our hypothesis, we conducted comparison experiments by adjusting the number of convolutional kernels to 4, 8, 16, and 32, respectively. The results showed that the loss function converged faster for the 16 and 32 kernels; moreover, the model's accuracy with the 16 kernel number increased steadily during the training process, and the final result was 99.53% better than the other kernel numbers (Figure 11). Therefore, our study used a model with the number of convolutional kernels set to 16 to obtain better performance.

Furthermore, we need to investigate how the network depth of the WTCN model affects the training results. We emphasize that since the sensory domain should cover all sequence data, when we change the number of network

layers, the kernel size of the corresponding convolutional kernels should be adjusted. During the training process, we found that the accuracy and precision indicators of the 5-layer and 7-layer networks fluctuated significantly; the 6-layer network was more stable, with an accuracy of 99.53% (Figure 12). Therefore, a 6-layer deep and 9-kernel size network model was used in our study.

5.4. Different Wavelets. So far, we have adjusted network parameters and structure to get better predictive accuracy. We should turn to select the optimal mother wavelet to process the raw data. Previously, we only used the Mexican Hat wavelet. In this part, we try to experiment with two more functions: Morlet and Gaus1. The Gaus1 and Mexican Hat wavelet shows identical good scores (Figure 13). Specifically, we analyze it and find that the Mexican Hat wavelet is the second derivative of the Gauss function, showing the same predictive ability as Gaus1. Therefore, we finally decided on the Mexican because of its better performance and improved accuracy.

5.5. Summarization of Results and Comparison with Previous Methods. In conclusion, compared to 1DCNN, LSTM, Hybrid, and TCN baseline networks, the WTCN model has achieved the best performance on the UniMib-SHAR

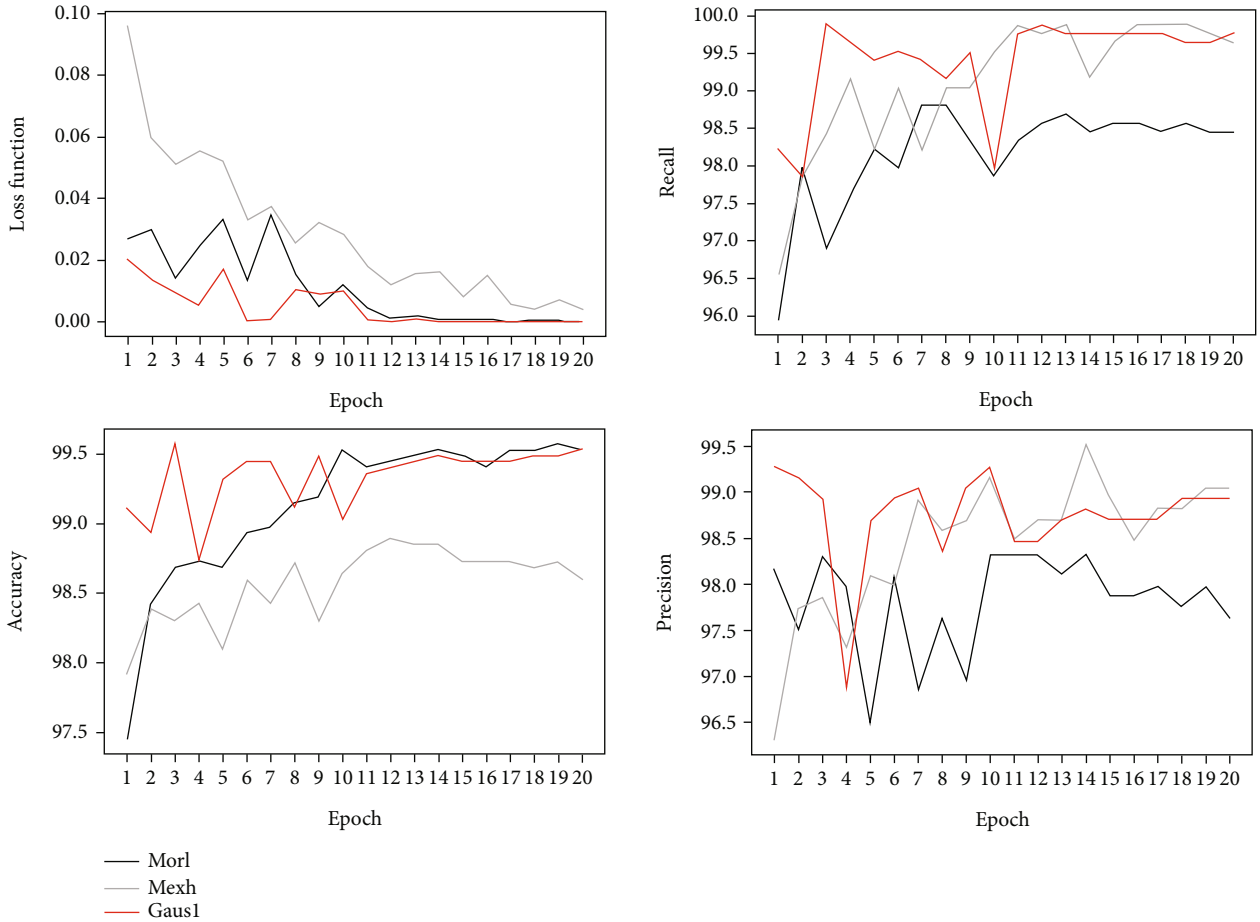


FIGURE 13: Loss function, recall, accuracy, and precision of number of epoches for different wavelet transforms of neural networks.

TABLE 5: Accuracy of WTCN model and other baseline models in experimental datasets.

Dataset method	WTCN	TCN	1DCNN	LSTM	Hybrid (CNN+LSTM)
UniMib-SHAR	99.53	99.28	98.90	96.18	99.11
SisFall-UMAFall	98.87	98.76	98.21	95.33	98.56

TABLE 6: Time of models.

Model	WTCN	TCN	1DCNN	LSTM	Hybrid (CNN+LSTM)
Time/s	0.0161	0.0156	0.0125	0.1373	0.0638

and SisFall-UMAFall datasets. Specifically, the accuracy was 99.53% on UniMib-SHAR and 98.87% on SisFall-UMAFall, respectively (Table 5).

In addition, our team also tested the computation time of the start-of-the-art model, and the results are shown in Table 6. We have found that WTCN was considerably faster than the LSTM model and slightly faster than the hybrid (CNN+LSTM) model, but slightly slower than 1DCNN and TCN. To analyze the reasons for this, firstly, as the 1DCNN model has fewer convolutional layers than the WTCN model, the amount of computation consumed during the experiments conducted by our team is less than that

of WTCN. Thus, the computation consumes slightly less time than WTCN. Secondly, as the WTCN model has an additional wavelet transform layer compared to the TCN model, it adds a little more computing time during the wavelet transform. In the future, we need to further focus on reducing the model's complexity while ensuring the prediction's accuracy, so as to lower the time delay when applying prediction models to practical applications.

Furthermore, Table 7 shows the results of the performance comparison between WTCN and other existing models. As previous papers tended to adopt a public dataset directly instead of integrating multiple datasets as we have

TABLE 7: Performance comparison (in %) of our WTCN model with other existing models.

Authors	Year	Model	Accuracy
Micucci et al. [14]	2017	SVM	98.71
Ivascu et al. [18]	2017	DNN	96.73
Mukherjee et al. [99]	2020	EnsemConvNet	99.69
Kanjilal et al. [100]	2021	1D-CNN feature	99.26
Kanjilal et al. [100]	2021	RNN-LSTM feature	99.45
Our team	2022	WTCN	99.53

done, our comparison and discussion are based on the UniMib-SHAR dataset only. On the whole, our WTCN outperforms almost all previous models. Moreover, it can be seen that the recognition accuracy of deep learning models (EnsemConvNet, 1D-CNN, RNN-LSTM, and WTCN) is better than that of the machine learning models (SVM and DNN) on fall detection tasks. Besides, it is worth noting that although EnsemConvNet has achieved excellent performance, it is a relatively complex model that includes CNN-Net, Encoded-Net, and CNN-LSTM. In order to achieve a tradeoff between accuracy and lightness, we prefer to design a light WTCN model, which would be suitable for installation on mobile phones or other wearable devices in the future.

6. Conclusions

Fall detection is one of the most challenging tasks in the human behavior recognition field. In order to solve the existing problems of CNN and RNN when they are used in these tasks, a well-performed temporal convolutional network (TCN) with wavelet transform has been proposed. The wavelet transform has been proved to be of the excellent capability to transform the raw signals from 1D to 2D without losing the details from raw signal data. Besides, because the TCN network has a deep causal convolution hierarchy and unique residual connection, it can deal with long sequences in time series data. By tuning parameters, we design a WTCN model with ultralong memory and stable gradients, which is capable of autoregression prediction. An experiment comparing the WTCN model with typical recursive architectures such as LSTM validates the robustness of the developed method.

Future work will extend in several directions. Firstly, there is a need to supplement realistic falls data for older age groups (>60 years) as much as possible. Specifically, the main problem of fall detection research is the difficulty of obtaining real falls data, as it is challenging to capture this type of data in the real-life setting of older people. However, it is necessary and meaningful to supplement this kind of data in order for the prediction model to work in real life. Secondly, given the complexity of real-life fall behavior occurrence, designing a robust prediction method that is insensitive to the conditions is vital for transforming fall detection from laboratory research into a practical application for health monitoring. In addition, while model-based and data-driven prediction methods can achieve a high degree of recognition accuracy, they also have limitations

such as generalization capabilities. Therefore, future research could also focus on hybrid models to explore the possibility of integrating different models by making full use of their strengths.

Data Availability

The original acceleration data supporting our research paper are from three public datasets (UniMiB SHAR, SisFall, and UMAFall), which have been cited in the main manuscript. In addition, processed data integrated by our team is also available. To assist future research, we have uploaded our integrated dataset at <https://www.kaggle.com/datasets/scoutofdan/fall-detection-dateset>.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This article was supported by the Fundamental Research Funds for the Central Universities in Chongqing University (2021CDSKXYTY003). The authors appreciate its support very much.

References

- [1] R. Liu, A. A. Ramli, H. Zhang, E. Henricson, and X. Liu, "An overview of human activity recognition using wearable sensors: healthcare and artificial intelligence," in *International Conference on Internet of Things*, Springer, Cham, 2021.
- [2] C. I. Tang, I. Perez-Pozuelo, D. Spathis, and C. Mascolo, "Exploring contrastive learning in human activity recognition for healthcare," 2020, <https://arxiv.org/abs/2011.11542>.
- [3] V. Bianchi, M. Bassoli, G. Lombardo, P. Fornacciari, M. Mordonini, and I. De Munari, "IoT wearable sensor and deep learning: an integrated approach for personalized human activity recognition in a smart home environment," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8553–8562, 2019.
- [4] E. Casilari, R. Lora-Rivera, and F. García-Lagos, "A study on the application of convolutional neural networks to fall detection evaluated with multiple public datasets," *Sensors*, vol. 20, no. 5, p. 1466, 2020.
- [5] World Health Organization (WHO) Falls (Facts sheet, 16), 2018, Available online: <https://www.who.int/news-room/fact-sheets/detail/falls> (accessed on 26 February 2020).

- [6] E. A. Vogels, *About One-In-Five Americans Use a Smart Watch or Fitness Tracker*, Pew Research Center, 2020.
- [7] R. Alharbi, T. Stump, N. Vafaie, A. Pfammatter, B. Spring, and N. Alshurafa, "I can't be myself: effects of wearable cameras on the capture of authentic behavior in the wild," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 3, pp. 1–40, 2018.
- [8] R. Alharbi, M. Tolba, L. C. Petito, J. Hester, and N. Alshurafa, "To mask or not to mask? Balancing privacy with visual confirmation utility in activity-oriented wearable cameras," *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies*, vol. 3, no. 3, pp. 1–29, 2019.
- [9] A. Hamid, A. Brahim, and O. Mohammed, "A survey of activity recognition in egocentric lifelogging datasets," in *2017 International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS)*, p. 8, Fez, Morocco, 2017.
- [10] M. S. Ryoo and L. Matthies, "First-person activity recognition: what are they doing to me?," in *Proceedings of the IEEE conference on computer vision and pattern recognition.*, Portland, OR, 2013.
- [11] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep learning for computer vision: a brief review," *Computational intelligence and neuroscience*, vol. 2018, Article ID 7068349, pp. 2730–2737, 2018.
- [12] S. Xu, J. Wang, W. Shou, T. Ngo, A.-M. Sadick, and X. Wang, "Computer vision techniques in construction: a critical review," *Archives of Computational Methods in Engineering*, vol. 28, no. 5, pp. 3383–3397, 2021.
- [13] N. Irvine, C. Nugent, S. Zhang, H. Wang, and W. W. Y. Ng, "Neural network ensembles for sensor-based human activity recognition within smart environments," *Sensors*, vol. 20, no. 1, p. 216, 2020.
- [14] D. Micucci, M. Mobilio, and P. Napolitano, "Unimib shar: a dataset for human activity recognition using acceleration data from smartphones," *Applied Sciences*, vol. 7, no. 10, p. 1101, 2017.
- [15] T. Ivascu, K. Cincar, A. Dinis, and V. Negru, "Activities of daily living and falls recognition and classification from the wearable sensors data," in *2017 E-Health and Bioengineering Conference (EHB)*, pp. 627–630, Sinaia, Romania, 2017.
- [16] C. A. Ronao and S.-B. Cho, "Deep convolutional neural networks for human activity recognition with smartphone sensors," in *International Conference on Neural Information Processing*, pp. 46–53, Istanbul, Turkey, 2015.
- [17] T. Zebin, P. J. Scully, and K. B. Ozanyan, "Human activity recognition with inertial sensors using a deep learning approach," in *2016 IEEE SENSORS*, p. 3, Orlando, FL, USA, 2016.
- [18] J. Yang, M. N. Nguyen, P. P. San, X. L. Li, and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition," in *Twenty-fourth international joint conference on artificial intelligence*, Buenos Aires, Argentina, 2015.
- [19] A. Ignatov, "Real-time human activity recognition from accelerometer data using convolutional neural networks," *Applied Soft Computing*, vol. 62, pp. 915–922, 2018.
- [20] Y. Guan and T. Plötz, "Ensembles of deep lstm learners for activity recognition using wearables," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 2, pp. 1–28, 2017.
- [21] Y. Zhao, R. Yang, G. Chevalier, X. Xu, and Z. Zhang, "Deep residual bidir-LSTM for human activity recognition using wearable sensors," *Mathematical Problems in Engineering*, vol. 2018, Article ID 7316954, 13 pages, 2018.
- [22] J. Chung, C. Gulcehre, K. H. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, <https://arxiv.org/abs/1412.3555>.
- [23] K. Czuszyński, J. Rumiński, and A. Kwaśniewska, "Gesture recognition with the linear optical sensor and recurrent neural networks," *IEEE Sensors Journal*, vol. 18, no. 13, pp. 5429–5438, 2018.
- [24] S. Mekruksavanich and A. Jitpattanakul, "RNN-based deep learning for physical activity recognition using smartwatch sensors: A case study of simple and complex activity recognition," *Mathematical Biosciences and Engineering*, vol. 19, no. 6, pp. 5671–5698, 2022.
- [25] O. Aziz, C. M. Russell, E. J. Park, and S. N. Robinovitch, "The effect of window size and lead time on pre-impact fall detection accuracy using support vector machine analysis of waist mounted inertial sensor data," in *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 30–33, Chicago, IL, USA, 2014.
- [26] N. N. Diep, C. Pham, and T. M. Phuong, "A classifier based approach to real-time fall detection using low-cost wearable sensors," in *2013 International Conference on Soft Computing and Pattern Recognition (SoCPar)*, pp. 14–20, Hanoi, Viet Nam, 2013.
- [27] B. T. Nukala, N. Shibuya, A. I. Rodriguez et al., "A real-time robust fall detection system using a wireless gait analysis sensor and an artificial neural network," in *2014 IEEE Healthcare Innovation Conference (HIC)*, pp. 219–222, Seattle, WA, USA, 2014.
- [28] H. Jian and H. Chen, "A portable fall detection and alerting system based on k-NN algorithm and remote medicine," *China Communications*, vol. 12, no. 4, pp. 23–31, 2015.
- [29] S.-H. Cheng, "An intelligent fall detection system using triaxial accelerometer integrated by active RFID," in *2014 International conference on machine learning and cybernetics*, pp. 517–522, Lanzhou, China, 2014.
- [30] G. Shi, J. Zhang, C. Dong, P. Han, Y. Jin, and J. Wang, "Fall detection system based on inertial mems sensors: analysis design and realization," in *2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, pp. 1834–1839, Shenyang, Peoples Republic of China, 2015.
- [31] A. Lisowska, G. Wheeler, V. Ceballos Inza, and I. Poole, "An evaluation of supervised, novelty-based and hybrid approaches to fall detection using Silmee accelerometer data," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 10–16, Santiago, CHILE, 2015.
- [32] D. Mrozek, A. Koczur, and B. Małysiak-Mrozek, "Fall detection in older adults with mobile IoT devices and machine learning in the cloud and on the edge," *Information Sciences*, vol. 537, pp. 132–147, 2020.
- [33] Y. Zheng, Q. Liu, E. Chen, Y. Ge, and J. L. Zhao, "Time series classification using multi-channels deep convolutional neural networks," *International conference on web-age information management*, 2014Springer, Cham, 2014.
- [34] Y. Zheng, Q. Liu, E. Chen, Y. Ge, and J. L. Zhao, "Exploiting multi-channels deep convolutional neural networks for multivariate time series classification," *Frontiers of Computer Science*, vol. 10, no. 1, pp. 96–112, 2016.

- [35] S. Matsui, N. Inoue, Y. Akagi, G. Nagino, and K. Shinoda, "User adaptation of convolutional neural network for human activity recognition," in *2017 25th European Signal Processing Conference (EUSIPCO)*, pp. 753–757, Kos, Greece, 2017.
- [36] S.-M. Lee, S. M. Yoon, and H. Cho, "Human activity recognition from accelerometer data using convolutional neural network," in *2017 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pp. 131–134, Jeju, South Korea, 2017.
- [37] K. Murakami and H. Taguchi, "Gesture recognition using recurrent neural networks," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 237–242, New Orleans, LA, USA, 1991.
- [38] A. Carfi, C. Motolese, B. Bruno, and F. Mastrogiovanni, "Online human gesture recognition using recurrent neural networks and wearable sensors," in *2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 188–195, Nanjing, China, 2018.
- [39] W. Chen, S. Wang, X. Zhang et al., "EEG-based motion intention recognition via multi-task RNNs," in *Proceedings of the 2018 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics*, pp. 279–287, San Diego, CA, USA, 2018.
- [40] M. Inoue, S. Inoue, and T. Nishida, "Deep recurrent neural network for mobile human activity recognition with high throughput," *Artificial Life and Robotics*, vol. 23, no. 2, pp. 173–185, 2018.
- [41] Y. Long, E. M. Jung, J. Kung, and S. Mukhopadhyay, "Reram crossbar based recurrent neural network for human activity detection," in *2016 international joint conference on neural networks (IJCNN)*, pp. 939–946, Vancouver, BC, Canada, 2016.
- [42] A. Tamamori, T. Hayashi, T. Toda, and K. Takeda, "An investigation of recurrent neural network for daily activity recognition using multi-modal signals," in *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 1334–1340, Kuala Lumpur, Malaysia, 2017.
- [43] A. Tamamori, T. Hayashi, T. Toda, and K. Takeda, "Daily activity recognition based on recurrent neural network using multi-modal signals," *APSIPA Transactions on Signal and Information Processing* 7, vol. 7, no. 1, 2018.
- [44] M. Z. Uddin, M. M. Hassan, A. Alsanad, and C. Savaglio, "A body sensor data fusion and deep recurrent neural network-based behavior recognition approach for robust healthcare," *Information Fusion*, vol. 55, pp. 105–115, 2020.
- [45] D.-W. Lee, K. Jun, K. Naheem, and M. S. Kim, "Deep neural network-based double-check method for fall detection using IMU-L sensor and RGB camera data," *IEEE Access*, vol. 9, pp. 48064–48079, 2021.
- [46] J. Ding and Y. Wang, "A WiFi-based smart home fall detection system using recurrent neural network," *IEEE Transactions on Consumer Electronics*, vol. 66, no. 4, pp. 308–317, 2020.
- [47] I. W. W. Wisesa and G. Mahardika, "Fall detection algorithm based on accelerometer and gyroscope sensor data using recurrent neural networks," in *IOP Conference Series: Earth and Environmental Science*, vol. 258, p. 10, Inst Teknologi Sumatera Campus, Lampung, INDONESIA, 2019.
- [48] E. Torti, A. Fontanella, M. Musci et al., "Embedded real-time fall detection with deep learning on wearable devices," in *2018 21st euromicro conference on digital system design (DSD)*, pp. 405–412, Prague, Czech Republic, 2018.
- [49] L. Zheng, S. Li, C. Zhu, and Y. Gao, "Application of IndRNN for human activity recognition: the Sussex-Huawei locomotion-transportation challenge," in *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers*, pp. 869–872, London, England, 2019.
- [50] G. Bailador, D. Roggen, G. Tröster, and G. Triviño, "Real time gesture recognition using continuous time recurrent neural networks," in *BodyNets*, p. 15, Florence, Italy, 2007.
- [51] T. Zebin, M. Sperrin, N. Peek, and A. J. Casson, "Human activity recognition from inertial sensor time-series using batch normalized deep LSTM recurrent networks," in *2018 40th annual international conference of the IEEE engineering in medicine and biology society (EMBC)*, pp. 4385–4388, Honolulu, HI, USA, 2018.
- [52] P. Khatiwada, M. Subedi, A. Chatterjee, and M. W. Gerdes, "Automated human activity recognition by colliding bodies optimization-based optimal feature selection with recurrent neural network," 2020, <https://arxiv.org/abs/2010.03324>.
- [53] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber, "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies," in *A Field Guide to Dynamical Recurrent Networks*, S. C. Kremer and J. F. Kolen, Eds., IEEE Press, Piscataway, NJ, USA, 2001.
- [54] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [55] A. Murad and J.-Y. Pyun, "Deep recurrent neural networks for human activity recognition," *Sensors*, vol. 17, no. 11, p. 2556, 2017.
- [56] P. Rivera, E. Valarezo, M. T. Choi, and T. S. Kim, "Recognition of human hand activities based on a single wrist IMU using recurrent neural networks," *International Journal of Pharma Medicine and Biological Sciences*, vol. 6, no. 4, pp. 114–118, 2017.
- [57] Y. Hu, X. Q. Zhang, L. Xu et al., "Harmonic loss function for sensor-based human activity recognition based on LSTM recurrent neural networks," *IEEE Access*, vol. 8, pp. 135617–135627, 2020.
- [58] F. J. Ordóñez and D. Roggen, "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, 2016.
- [59] W. Ahmad, B. M. Kazmi, and H. Ali, "Human activity recognition using multi-head CNN followed by LSTM," in *2019 15th international conference on emerging technologies (ICET)*, p. 6, Peshawar, Pakistan, 2019.
- [60] S. Deep and X. Zheng, "Hybrid model featuring CNN and LSTM architecture for human activity recognition on smartphone sensor data," in *2019 20th international conference on parallel and distributed computing, applications and technologies (PDCAT)*, pp. 259–264, Gold Coast, QLD, Australia, 2019.
- [61] B. Friedrich, C. Lübke, and A. Hein, "Combining LSTM and CNN for mode of transportation classification from smartphone sensors," in *Adjunct Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers*, pp. 305–310, Mexico, 2020.

- [62] J. Gao, P. Gu, Q. Ren, J. Zhang, and X. Song, "Abnormal gait recognition algorithm based on LSTM-CNN fusion network," *IEEE Access*, vol. 7, pp. 163180–163190, 2019.
- [63] R. Mutegeki and D. S. Han, "A CNN-LSTM approach to human activity recognition," in *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIIC)*, pp. 362–366, Fukuoka, Japan, 2020.
- [64] V. Y. Senyurek, M. H. Imtiaz, P. Belsare, S. Tiffany, and E. Sazonov, "A CNN-LSTM neural network for recognition of puffing in smoking episodes using wearable sensors," *Biomedical Engineering Letters*, vol. 10, no. 2, pp. 195–203, 2020.
- [65] Y. Wu, B. Zheng, and Y. Zhao, "Dynamic gesture recognition based on LSTM-CNN," in *2018 Chinese Automation Congress (CAC)*, pp. 2446–2450, Xi'an, China, 2018.
- [66] K. Xia, J. Huang, and H. Wang, "LSTM-CNN architecture for human activity recognition," *IEEE Access*, vol. 8, pp. 56855–56866, 2020.
- [67] S. Bai, J. Zico Kolter, and V. Koltun, "Trellis networks for sequence modeling," 2018, <https://arxiv.org/abs/1810.06682>.
- [68] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," 2018, <https://arxiv.org/abs/1803.01271>.
- [69] R. A. Hamad, M. Kimura, L. Yang, W. L. Woo, and B. Wei, "Dilated causal convolution with multi-head self attention for sensor human activity recognition," *Neural Computing and Applications*, vol. 33, no. 20, pp. 13705–13722, 2021.
- [70] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," 2015, <https://arxiv.org/abs/1511.07122>.
- [71] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, Las Vegas, Nevada, USA, 2016.
- [72] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Thirty-first AAAI conference on artificial intelligence*, vol. 31, pp. 4278–4284, San Francisco, CA, USA, 2017.
- [73] S. R. Park and J. Lee, "A fully convolutional neural network for speech enhancement," 2016, <https://arxiv.org/abs/1609.07132>.
- [74] M. Sifuzzaman, M. Rafiq Islam, and M. Z. Ali, "Application of wavelet transform and its advantages compared to fourier transform," *Journal of Physical Sciences*, vol. 13, pp. 121–134, 2009.
- [75] I. Daubechies and B. J. Bates, "Ten lectures on wavelets," *The Journal of the Acoustical Society of America*, vol. 93, no. 3, p. 1671, 1993.
- [76] P. M. Bentley and J. T. E. McDonnell, "Wavelet transforms: an introduction," *Electronics & Communication Engineering Journal*, vol. 6, no. 4, pp. 175–186, 1994.
- [77] M. Farge, "Wavelet transforms and their applications to turbulence," *Annual Review of Fluid Mechanics*, vol. 24, no. 1, pp. 395–458, 1992.
- [78] O. Rioul and P. Duhamel, "Fast algorithms for discrete and continuous wavelet transforms," *IEEE Transactions on Information Theory*, vol. 38, no. 2, pp. 569–586, 1992.
- [79] J.-P. Antoine, P. Carrette, R. Murenzi, and B. Piette, "Image analysis with two-dimensional continuous wavelet transform," *Signal Processing*, vol. 31, no. 3, pp. 241–272, 1993.
- [80] R. S. Stanković and B. J. Falkowski, "The Haar wavelet transform: its status and achievements," *Computers & Electrical Engineering*, vol. 29, no. 1, pp. 25–44, 2003.
- [81] J. Lin and Q. Liangsheng, "Feature extraction based on Morlet wavelet and its application for mechanical fault diagnosis," *Journal of Sound and Vibration*, vol. 234, no. 1, pp. 135–148, 2000.
- [82] S. Z. Mahmoodabadi, A. Ahmadian, and M. D. Abolhasani, "ECG feature extraction using Daubechies wavelets," in *Proceedings of the fifth IASTED International conference on Visualization, Imaging and Image Processing*, pp. 343–348, Benidorm, SPAIN, 2005.
- [83] S.-J. Huang and C.-T. Hsieh, "Coiflet wavelet transform applied to inspect power system disturbance-generated signals," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 38, no. 1, pp. 204–210, 2002.
- [84] T. Hou and H. Qin, "Continuous and discrete Mexican hat wavelet transforms on manifolds," *Graphical Models*, vol. 74, no. 4, pp. 221–232, 2012.
- [85] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: a local SVM approach," in *Proceedings of the 17th International Conference on Pattern Recognition, 2004*, vol. 3, British Machine Vis Assoc, Cambridge, ENGLAND, 2004ICPR 2004.
- [86] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 12, pp. 2247–2253, 2007.
- [87] T. L. van Kasteren, G. Englebienne, and B. J. Kröse, "Human activity recognition from wireless sensor network data: benchmark and software," in *Activity Recognition in Pervasive Intelligent Environments*, pp. 165–186, Atlantis Press, 2011.
- [88] A. Saeed, T. Ozcelebi, and J. Lukkien, "Multi-task self-supervised learning for human activity detection," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 3, no. 2, pp. 1–30, 2019.
- [89] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *ACM SIGKDD Explorations Newsletter*, vol. 12, no. 2, pp. 74–82, 2011.
- [90] D. Anguita, A. Ghio, L. Oneto, X. Parra Perez, and J. L. Reyes Ortiz, "A public domain dataset for human activity recognition using smartphones," in *Proceedings of the 21th international European symposium on artificial neural networks, computational intelligence and machine learning*, 2013.
- [91] A. Sucerquia, J. López, and J. Vargas-Bonilla, "SisFall: a fall and movement dataset," *Sensors*, vol. 17, no. 12, p. 198, 2017.
- [92] E. Casilari, J. A. Santoyo-Ramón, and J. M. Cano-García, "Analysis of a smartphone-based architecture with multiple mobility sensors for fall detection," *PLoS One*, vol. 11, no. 12, article e0168069, 2016.
- [93] A. Nedorubova, A. Kadyrova, and A. Khlyupin, "Human activity recognition using continuous wavelet transform and convolutional neural networks," 2021, <https://arxiv.org/abs/2106.12666>.
- [94] X. Cheng, L. Zhang, Y. Tang, Y. Liu, H. Wu, and J. He, "Real-time human activity recognition using conditionally parameterized convolutions on mobile and wearable devices," *IEEE Sensors Journal*, vol. 22, no. 6, pp. 5889–5901, 2022.
- [95] Y. Tang, Q. Teng, L. Zhang, F. Min, and J. He, "Layer-wise training convolutional neural networks with smaller filters

- for human activity recognition using wearable sensors,” *IEEE Sensors Journal*, vol. 21, no. 1, pp. 581–592, 2020.
- [96] G. Vavoulas, C. Chatzaki, T. Malliotakis, M. Pediaditis, and M. Tsiknakis, “The mobiact dataset: recognition of activities of daily living using smartphones,” in *International Conference on Information and Communication Technologies for Ageing Well and e-Health*, vol. 2, pp. 143–151, Rome, ITALY, 2016.
- [97] A. L. Maas, A. Y. Hannun, and A. Y. Ng, “Rectifier nonlinearities improve neural network acoustic models,” in *Proc. icml*, vol. 30, p. 6, Stanford University, CA, USA, 2013.
- [98] D. Mukherjee, R. Mondal, P. K. Singh, R. Sarkar, and D. Bhattacharjee, “EnsemConvNet: a deep learning approach for human activity recognition using smartphone sensors for healthcare applications,” *Multimedia Tools and Applications*, vol. 79, no. 41–42, pp. 31663–31690, 2020.
- [99] R. Kanjilal and I. Uysal, “The future of human activity recognition: deep learning or feature engineering?,” *Neural Processing Letters*, vol. 53, no. 1, pp. 561–579, 2021.
- [100] G. Vavoulas, M. Pediaditis, C. Chatzaki, E. G. Spanakis, and M. Tsiknakis, “The mobifall dataset: fall detection and classification with a smartphone,” *International Journal of Monitoring and Surveillance Technologies Research (IJMSTR)*, vol. 2, no. 1, pp. 44–56, 2014.